

# Check whether various gene signatures predict cisplatin response

November 16, 2008

## 1 Load and prepare data

### 1.1 Load add-on packages

```
> library(affy)
> library(hgu133plus2.db)
> library(mvtnorm)
```

### 1.2 Load the RMA-processed expression values

Using manufacturer-defined probe sets:

```
> load("~/aron/sbge_cancer/data/exprs/platinum.rma.RData")
```

Using redefined probesets based on Aceview:

```
> load("~/aron/sbge_cancer/data/exprs/platinum.arma.RData")
```

### 1.3 Remove outliers and non-cisplatin-treated samples

```
> ok.sample <- !is.na(platinum.rma$Miller)
> ok.sample[platinum.rma$Chip %in% c("P9", "P20")] <- FALSE
> plat.rma <- platinum.rma[, ok.sample]
> plat.arma <- platinum.arma[, ok.sample]
```

### 1.4 Define the response groups

We are interested in the Miller-Payne response score. Note: MP score = 0 means "progression". We are also interested in the distinction between two response groups: Miller-Payne grades 0-2, and M-P grades 3-5.

```
> MP.score <- plat.rma$Miller.Payne.Grade
> binary.response <- factor(ifelse(MP.score > 2, "MP345", "MP012"))
> table(binary.response, MP.score)

      MP.score
binary.response 0 1 2 3 4 5
                  MP012 3 5 4 0 0 0
                  MP345 0 0 0 4 4 4
```

## 1.5 Load the signature genes

Probeset-indexed signatures from Bild et al (2006). Downloaded June 21, 2007 from  
<http://data.cgt.duke.edu/oncogene.php>

```
> load("~/aron/chb/data/int/sigs.bild.RData")
> str(sigs.bild)

List of 5
$ betacatenin:'data.frame':      98 obs. of  2 variables:
 ..$ probe: chr [1:98] "225098_at" "218150_at" "222667_s_at" "208859_s_at" ...
 ..$ ratio: num [1:98] 0.853 0.869 0.725 0.783 6.454 ...
$ e2f3     :'data.frame':      298 obs. of  2 variables:
 ..$ probe: chr [1:298] "223320_s_at" "213485_s_at" "209735_at" "239579_at" ...
 ..$ ratio: num [1:298] 1.85 0.66 3.59 3.73 1.66 ...
$ myc      :'data.frame':      248 obs. of  2 variables:
 ..$ probe: chr [1:248] "208161_s_at" "209641_s_at" "231907_at" "234312_s_at" ...
 ..$ ratio: num [1:248] 0.619 0.583 0.808 0.777 0.690 ...
$ ras      :'data.frame':      348 obs. of  2 variables:
 ..$ probe: chr [1:348] "203504_s_at" "205179_s_at" "205180_s_at" "219935_at" ...
 ..$ ratio: num [1:348] 0.331 5.658 3.848 0.206 3.487 ...
$ src      :'data.frame':      73 obs. of  2 variables:
 ..$ probe: chr [1:73] "213485_s_at" "201128_s_at" "215867_x_at" "201879_at" ...
 ..$ ratio: num [1:73] 0.689 0.587 0.643 0.902 0.660 ...
```

Our gene-indexed collection of signatures:

```
> load("~/aron/chb/data/int/sigs_SC_ACE-hg17.RData")
```

Define the subset of signatures we are interested in:

```
> sigs.gene <- sigs[c("CC", "CIN70", "CSR")]
> str(sigs.gene)
```

```
List of 3
$ CC   :'data.frame':      764 obs. of  2 variables:
 ..$ gene : chr [1:764] "ACD" "ACYP1" "ADAMTS1" "ADCK2" ...
 ..$ weight: int [1:764] 1 1 1 1 1 1 1 1 1 ...
$ CIN70:'data.frame':      70 obs. of  2 variables:
 ..$ gene : chr [1:70] "TPX2" "PRC1" "FOXM1" "CDC2" ...
 ..$ weight: int [1:70] 1 1 1 1 1 1 1 1 1 ...
$ CSR   :'data.frame':      402 obs. of  2 variables:
 ..$ gene : chr [1:402] "ADAMTS1" "AND-1" "ARHC" "BAF53A" ...
 ..$ weight: int [1:402] 1 1 1 1 1 1 1 1 1 ...
```

"CC" is the set of cell-cycle genes described by ML Whitfield et al., 2002. "CIN70" is a signature of chromosomal instability described by SL Carter et al., 2006. "CSR" is the Core Serum Response described by HY Chang et al., 2004.

## 1.6 Miscellaneous

We will need gene names and symbols for each probe set.

```
> symb <- sapply(mget(featureNames(plat.rma), hgu133plus2SYMBOL),
+   function(x) x[1])
> gene <- sapply(mget(featureNames(plat.rma), hgu133plus2GENENAME),
+   function(x) x[1])
```

A convenient plotting function:

```
> pl <- function(x, main) {
+   par(mfrow = c(3, 5), mar = c(5, 4, 1, 2), oma = c(0, 0, 2,
+     0))
+   layout(matrix(1:2, nrow = 1), widths = c(1, 2))
+   p <- t.test(x ~ binary.response)$p.value
+   p.txt <- paste("P =", format(p, digits = 2, scientific = FALSE))
+   corr <- cor.test(x, MP.score)
+   corr.txt <- paste("cor =", format(corr$estimate, digits = 2,
+     scientific = FALSE), "; p =", format(corr$p.value, digits = 2,
+     scientific = FALSE))
+   ylim <- range(x)
+   ylim <- c(ylim[1], ylim[2] + (diff(ylim) * 0.1))
+   boxplot(x ~ binary.response, ylim = ylim)
+   legend("top", legend = p.txt, bty = "n")
+   stripchart(x ~ MP.score, ylim = ylim, method = "jitter",
+     vertical = TRUE, xlab = "MP score", ylab = "")
+   legend("top", legend = corr.txt, bty = "n")
+   title(outer = TRUE, main = main)
+ }
```

## 2 Calculation of signature scores

### 2.1 Probeset-based signatures (Bild)

```
> plat.bild <- sapply(sigs.bild, function(x) {
+   expr <- exprs(plat.rma)[x$probe, ]
+   weight <- sign(log2(x$ratio))
+   scale(colMeans(expr * weight), scale = FALSE)
+ })
```

### 2.2 Gene-based signatures (CC, CIN, CSR)

```
> plat.sigs <- sapply(sigs.gene, function(sig) {
+   sig <- sig[sig$gene %in% featureNames(plat.arma), ]
+   expr <- exprs(plat.arma)[sig$gene, ]
+   scale(colMeans(expr * sig$weight), scale = FALSE)
+ })
```

### 3 Are any signatures associated with cisplatin response?

```
> all.sigs <- cbind(plat.bild, plat.sigs)
```

#### 3.1 t tests against binary response

```
> ttest.pvalue <- apply(all.sigs, 2, function(x) t.test(x ~ binary.response)$p.value)

> data.frame(p.value = sort(ttest.pvalue))
```

	p.value
e2f3	0.0791365
src	0.1952881
CSR	0.2386784
CC	0.2448897
betacatenin	0.2465621
ras	0.2865095
CIN70	0.3288609
myc	0.4328322

None of these achieve statistically significance.

#### 3.2 Correlation tests using Miller-Payne score

```
> cortests.pvalue <- apply(all.sigs, 2, function(x) cor.test(x,
+           MP.score)$p.value)

> data.frame(p.value = sort(cortests.pvalue))
```

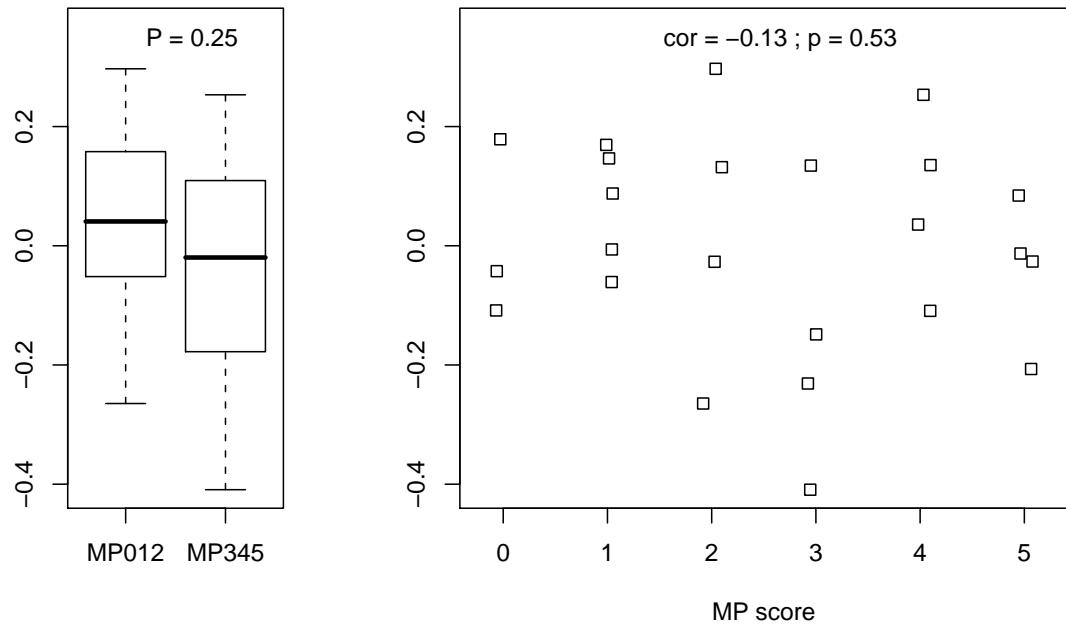
	p.value
e2f3	0.02520256
CSR	0.09798359
CC	0.10849958
CIN70	0.12794075
myc	0.21381474
src	0.42097669
betacatenin	0.53372045
ras	0.71139699

The only signature reaching statistical significance is E2F3.

## 4 Figures for each signature

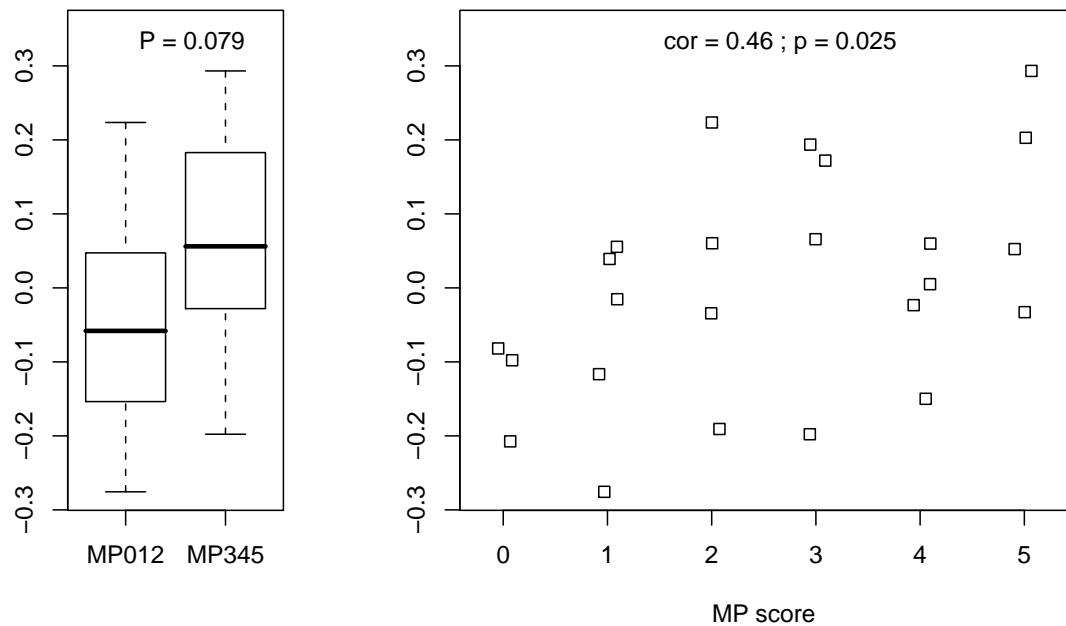
```
> pl(plat.bild[, "betacatenin"], main = "Bild: beta-catenin")
```

**Bild: beta-catenin**

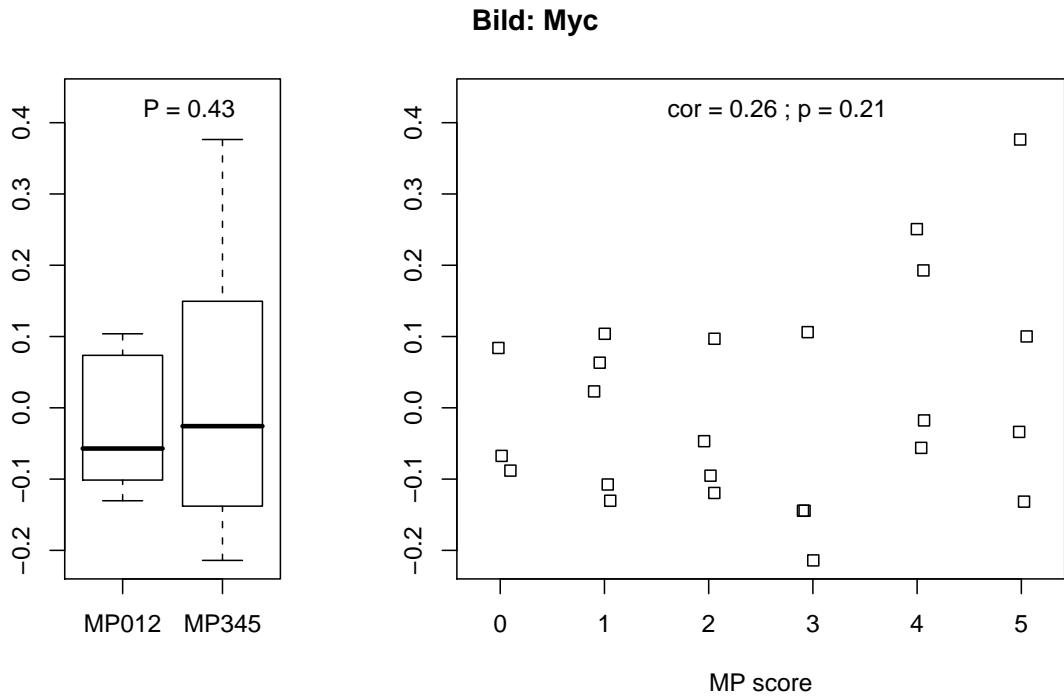


```
> pl(plat.bild[, "e2f3"], main = "Bild: E2F3")
```

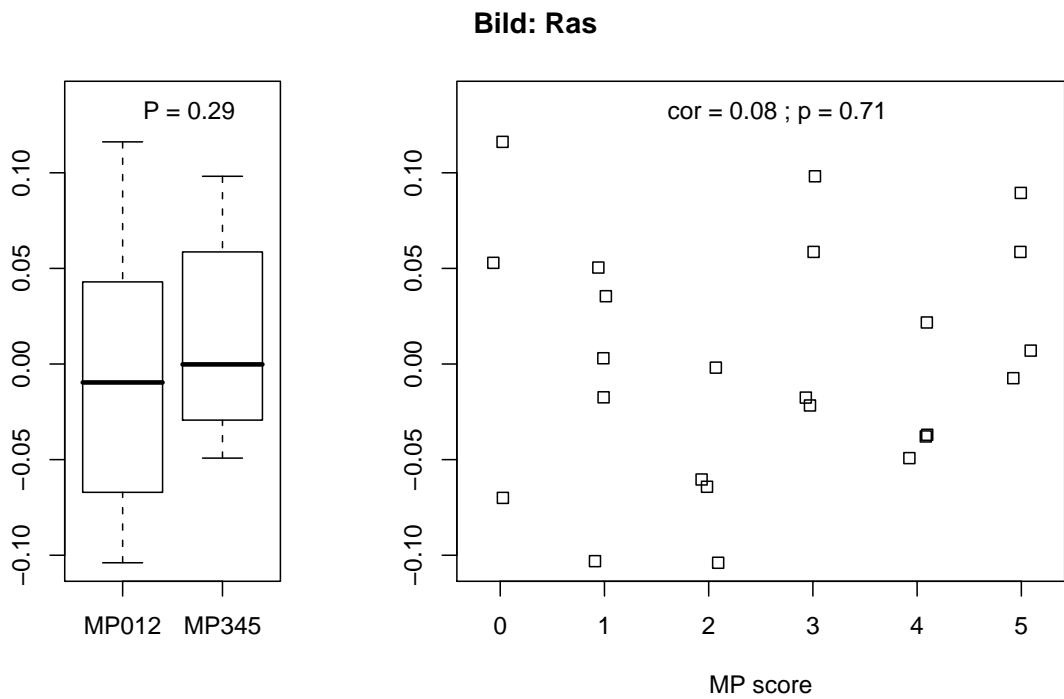
**Bild: E2F3**



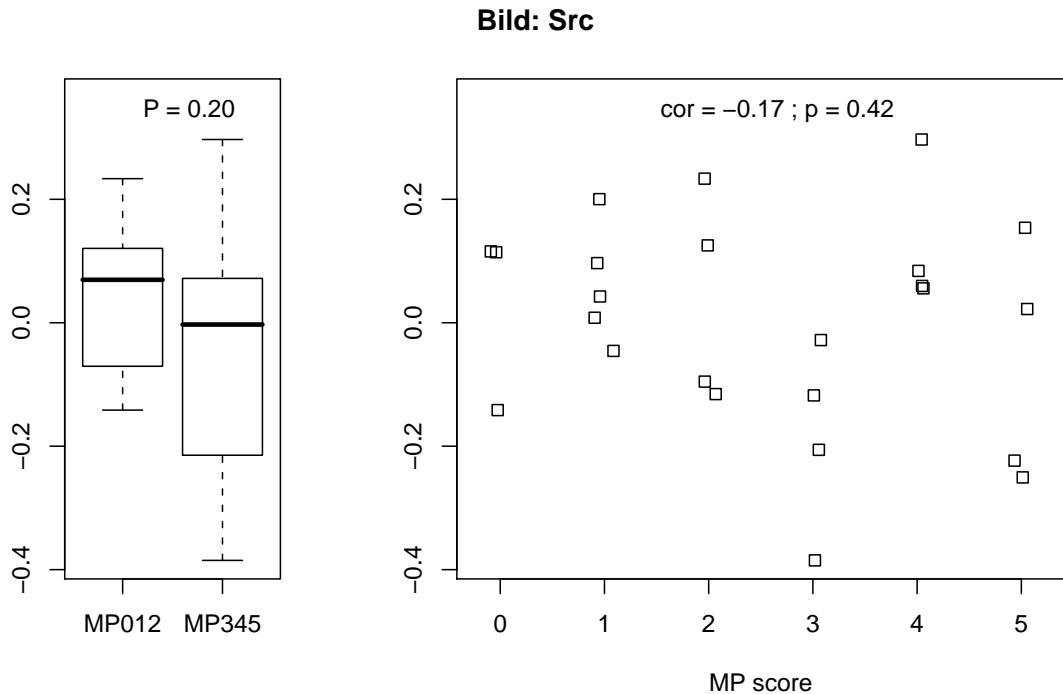
```
> pl(plat.bild[, "myc"], main = "Bild: Myc")
```



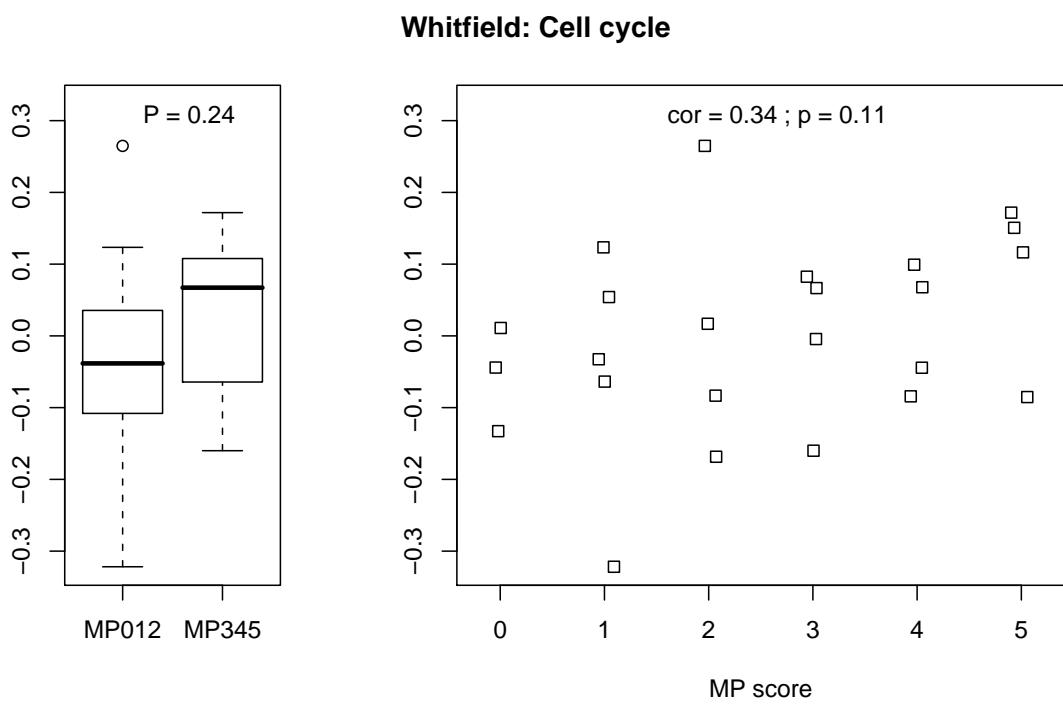
```
> pl(plat.bild[, "ras"], main = "Bild: Ras")
```



```
> pl(plat.bild[, "src"], main = "Bild: Src")
```

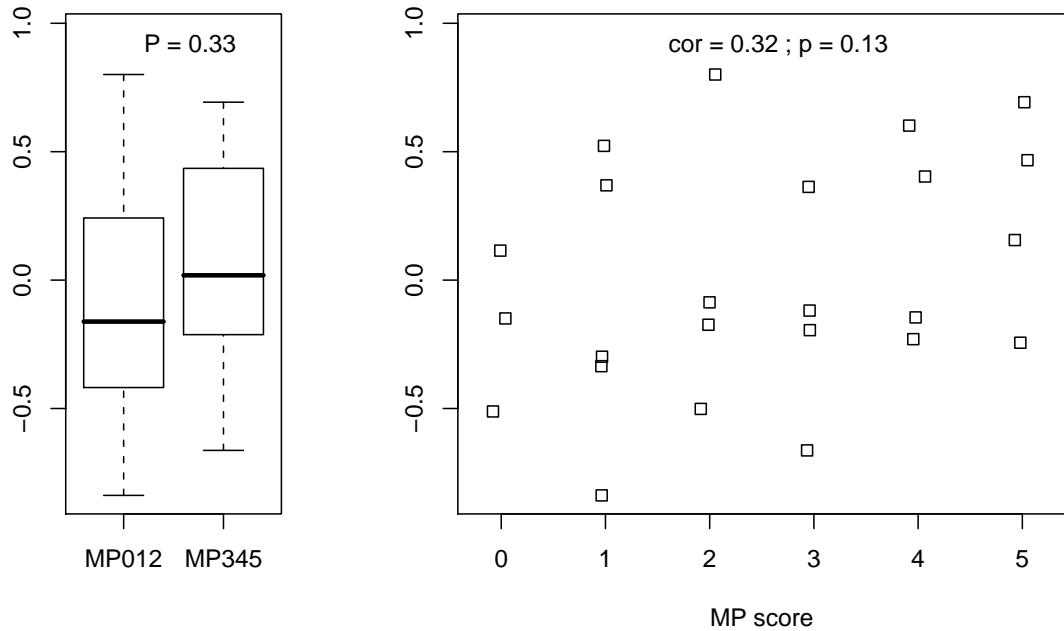


```
> pl(plat.sigs[, "CC"], main = "Whitfield: Cell cycle")
```



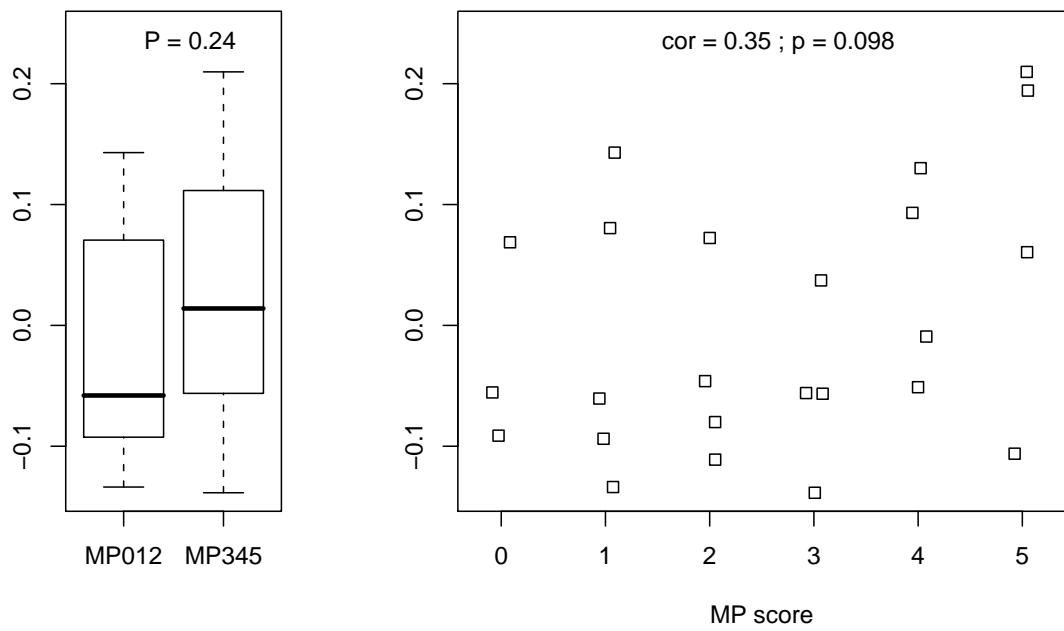
```
> pl(plat.sigs[, "CIN70"], main = "Carter: Chromosomal instability")
```

**Carter: Chromosomal instability**



```
> pl(plat.sigs[, "CSR"], main = "Chang: Serum response")
```

**Chang: Serum response**



## 5 Immune response classifier

As described in Teschendorff and Caldas, 2008, Breast Cancer Research.

### 5.1 Define constants and functions

These are the constants from Table 1:

```
> class.names <- c("good.down", "good.up", "poor.down", "poor.up")
> gene.symbols <- c("HLA-F", "IGLC2", "LY9", "TNFRSF17", "SPP1",
+   "XCL2", "C1QA")
> mu.hat <- matrix(c(-0.31, 0.65, -0.29, 0.4, -0.56, 0.98, -0.46,
+   0.68, -0.29, 0.58, -0.52, 1.12, -0.41, 0.97, -0.58, 0.59,
+   0.01, -0.38, 0.47, -0.57, -0.36, 0.67, -0.41, 0.58, -0.39,
+   -0.79, -0.4, 0.57), nrow = 7, byrow = TRUE)
> dimnames(mu.hat) <- list(gene.symbols, class.names)
> Sigma.hat <- c(0.74, 0.74, 0.58, 0.58)
> names(Sigma.hat) <- class.names
> pi.hat <- c(0.31, 0.28, 0.32, 0.09)
> names(pi.hat) <- class.names
> rbind(mu.hat, Sigma.hat, pi.hat)
```

	good.down	good.up	poor.down	poor.up
HLA-F	-0.31	0.65	-0.29	0.40
IGLC2	-0.56	0.98	-0.46	0.68
LY9	-0.29	0.58	-0.52	1.12
TNFRSF17	-0.41	0.97	-0.58	0.59
SPP1	0.01	-0.38	0.47	-0.57
XCL2	-0.36	0.67	-0.41	0.58
C1QA	-0.39	-0.79	-0.40	0.57
Sigma.hat	0.74	0.74	0.58	0.58
pi.hat	0.31	0.28	0.32	0.09

Here we implement the classifier from Equation 5. The following function expects a vector of length 7, with the genes in the same order as given above, with NA for any unavailable genes.

```
> classify <- function(y) {
+   require(mvtnorm)
+   ok <- !is.na(y)
+   scores <- sapply(class.names, function(i) {
+     sigma <- diag(sum(ok)) * Sigma.hat[i]
+     pi.hat[i] * dmvnorm(y[ok], mean = mu.hat[ok, i], sigma = sigma)
+   })
+   names(scores) <- class.names
+   scores/sum(scores)
+ }
```

## 5.2 Apply the classifier to the cisplatin trial data

Get expression values for the genes of interest. If multiple probe sets interrogate a gene of interest, use their average. The gene "IGLC2" is not found using our annotation, but Ensembl maps two probe sets to IGLC2, so we will use these.

```
> all.pr <- sapply(gene.symbols, function(x) names(which(symb ==
+      x)))
> all.pr[["IGLC2"]] <- c("209138_x_at", "214677_x_at")
> all.pr

$`HLA-F`
[1] "204806_x_at" "221875_x_at" "221978_at"

$IGLC2
[1] "209138_x_at" "214677_x_at"

$LY9
[1] "210370_s_at" "215967_s_at" "231124_x_at"

$TNFRSF17
[1] "206641_at"

$SPP1
[1] "1568574_x_at" "209875_s_at"

$XCL2
[1] "214567_s_at"

$C1QA
[1] "218232_at"

> plat.imm <- t(sapply(all.pr, function(x) colMeans(exprs(plat.rma))[x,
+      , drop = FALSE]))
> str(plat.imm)

num [1:7, 1:24] 5.60 6.22 4.36 3.52 5.05 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:7] "HLA-F" "IGLC2" "LY9" "TNFRSF17" ...
..$ : chr [1:24] "AR2006110802.CEL" "AR2006110803.CEL" "AR2006110804.CEL" "AR2006110805.CEL"
```

Center and scale (Z-transform) the data and apply the classifier.

```
> plat.imm.scaled <- t(scale(t(plat.imm)))
> plat.imm.scores <- t(apply(plat.imm.scaled, 2, classify))
> plat.imm.class <- class.names[apply(plat.imm.scores, 1, which.max)]
> table(plat.imm.class)

plat.imm.class
good.down   good.up poor.down   poor.up
      8          3         9          4
```

### 5.3 Does the classifier predict response?

```
> table(plat.imm.class, binary.response)
```

```
              binary.response
plat.imm.class MP012 MP345
  good.down     4     4
  good.up       2     1
  poor.down    4     5
  poor.up      2     2
```

```
> table(plat.imm.class, MP.score)
```

```
              MP.score
plat.imm.class 0 1 2 3 4 5
  good.down 0 2 2 2 1 1
  good.up   2 0 0 0 0 1
  poor.down 1 2 1 2 1 2
  poor.up    0 1 1 0 2 0
```

The classifier does not seem to predict response.

## 6 sessionInfo

The results in this file are generated using the following packages:

```
> sessionInfo()

R version 2.7.2 (2008-08-25)
i386-apple-darwin8.11.1

locale:
C

attached base packages:
[1] tools      stats       graphics   grDevices  utils      datasets   methods
[8] base

other attached packages:
[1] mvtnorm_0.9-2        hgu133plus2.db_2.2.0 AnnotationDbi_1.2.2
[4] RSQLite_0.6-9         DBI_0.2-4            affy_1.18.2
[7] preprocessCore_1.2.0  affyio_1.8.0       Biobase_2.0.1

> system("uname -a", intern = TRUE)

[1] "Darwin eklund.local 9.5.0 Darwin Kernel Version 9.5.0: Wed Sep 3 11:29:43 PDT 2008;
```